

Concurrent I/O Management for Cluster-based I/O Storage



Kai Shen

Dept. of Computer Science, Univ. of Rochester

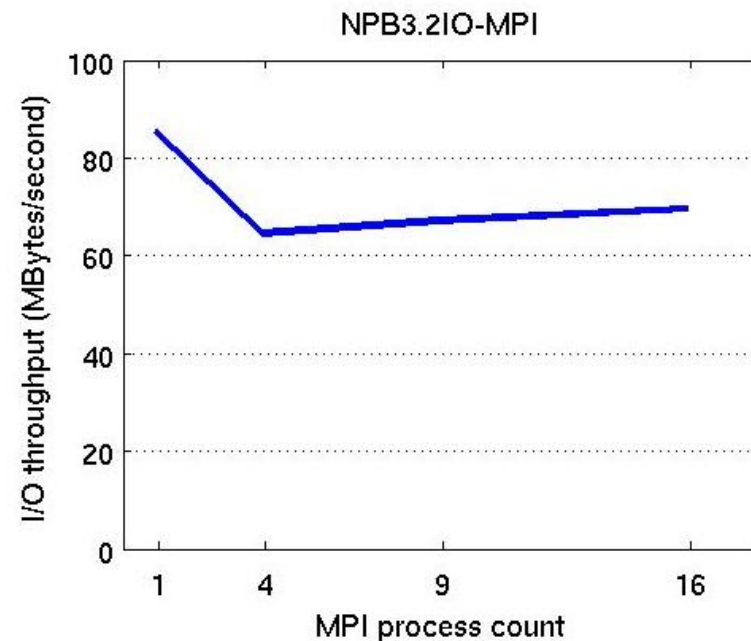
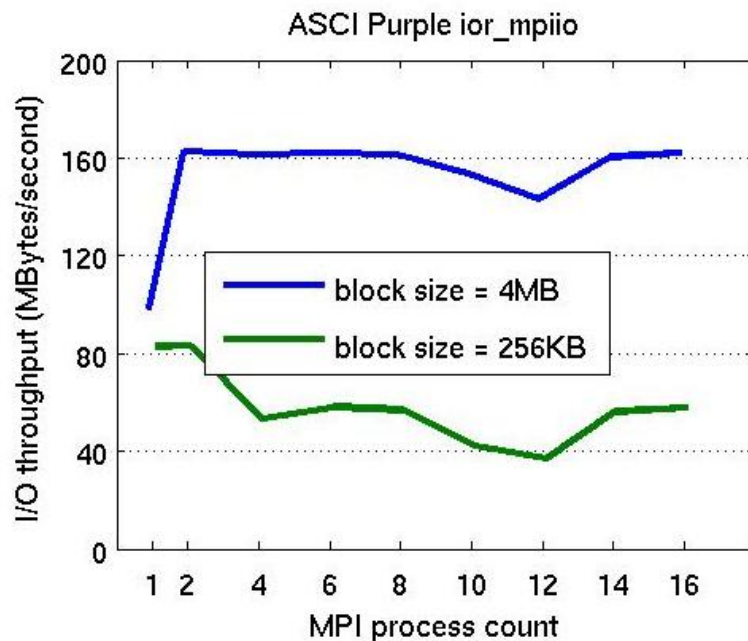


Project Overview

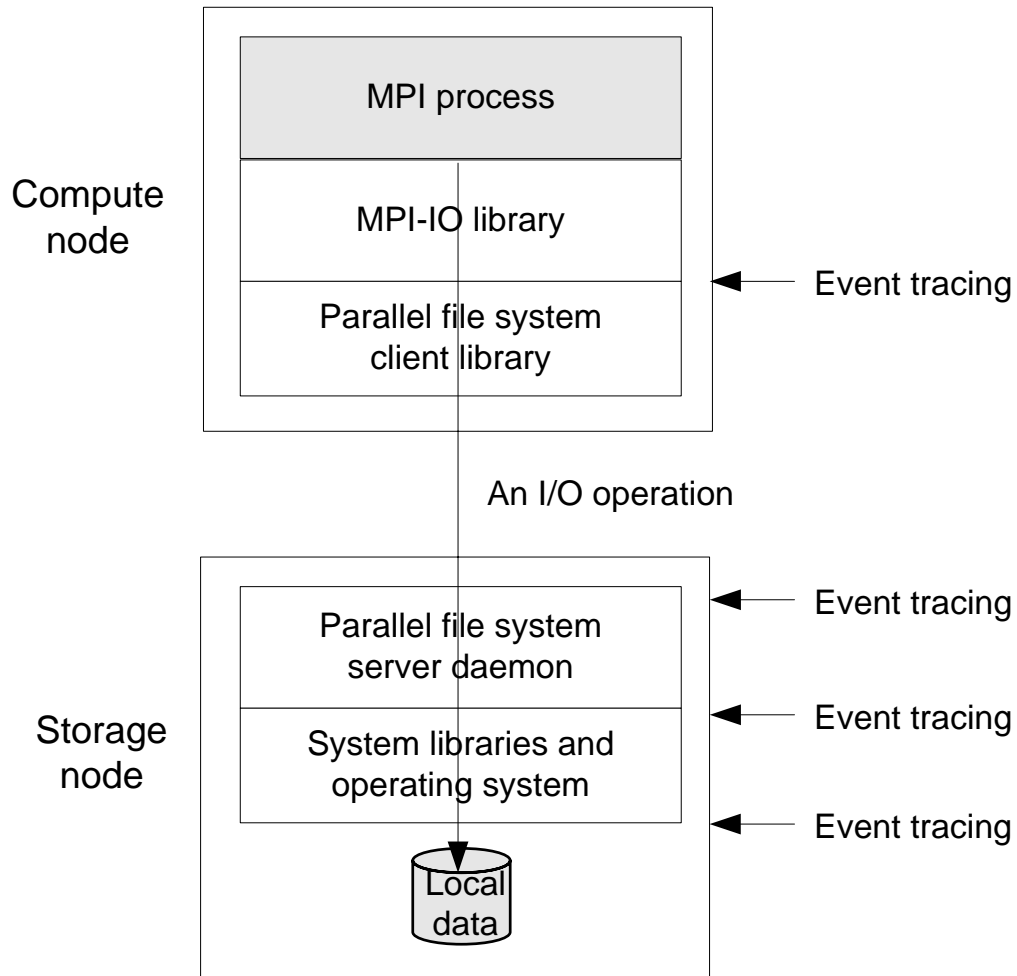
- Data-intensive high-end applications demand scalable I/O capacity
- Achieving reliably high performance is difficult since parallel I/O systems are complex
 - computing parallelism combined with I/O concurrency
 - multiple layers of system components – deep I/O stack
- Research goals:
 - develop systematic approach to analyze performance problems and identify causes (focus on concurrency-related issues)
 - devise new system-level techniques to address discovered problems

A Quantitative Example (I/O Read Throughput)

- Up to 16 compute nodes running MPICH2 (MPI-IO)
- 6 striped storage nodes running PVFS2; each run Linux 2.6.12
- Gigabit Ethernet (~80us TCP/IP roundtrip latency)



Multi-layer Event Tracing and Analysis



- Many layers of software present possible sources of problems
- Derive generic (layer-independent) I/O characteristics to understand cross-layer evolution.
- Perform bottom-up trace analysis (relatively easy anomaly identification at lower system layers).



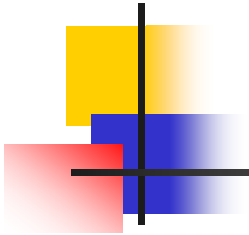
Results of Trace Analysis

- Problem #1: interleaved I/O under concurrent operations.
- Cause: insufficient I/O prefetching.
- Problem #2: slow return of I/O that should hit the cache.
- Cause: C library serializes AIOs on one file description.
- Problem #3: long delay in asynchronous write calls.
- Cause: client block on server-side write buffering.



Motivation for Next Step Work

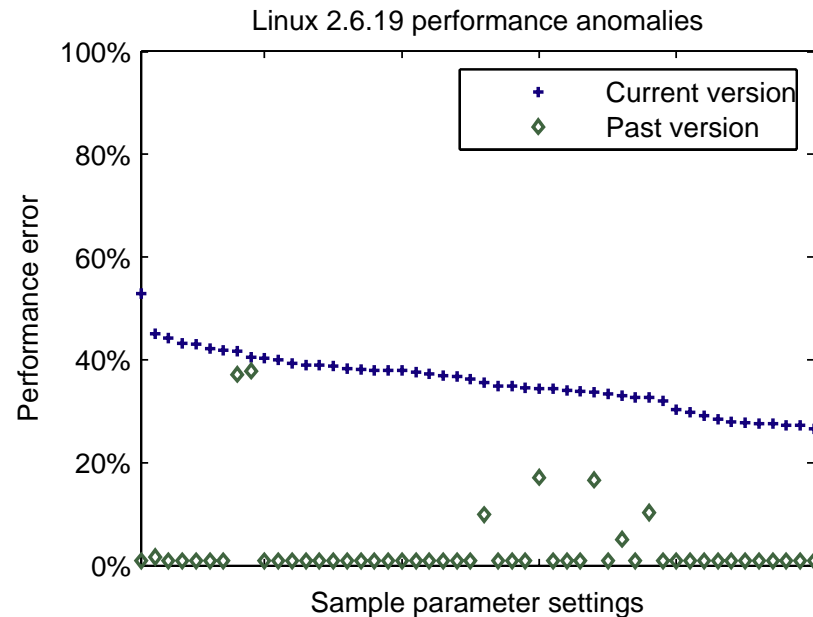
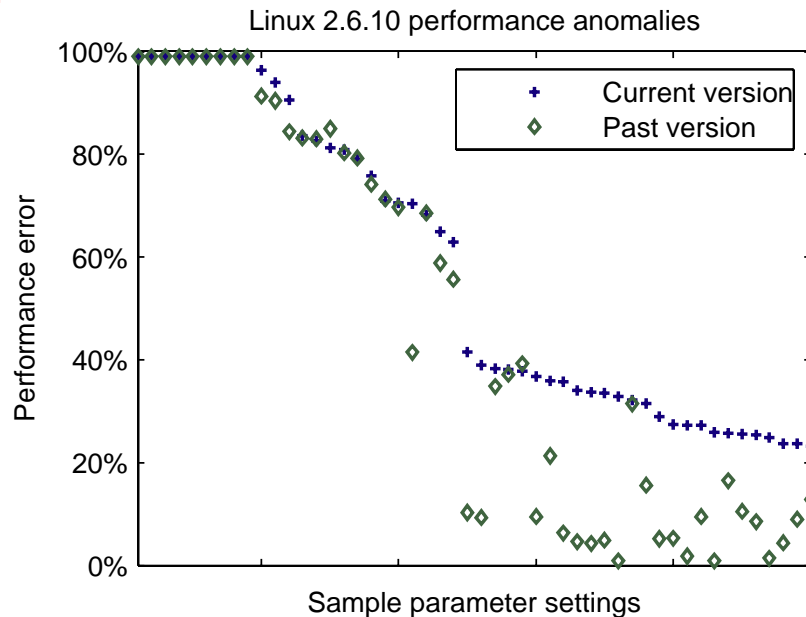
- Trace analysis for performance anomaly detection
 - Event trace analysis is still largely manual.
 - Dilemma:
 - too many traced events are difficult for human analysis
 - tracing too events few may miss important anomalies/errors
 - Wish for:
 - tracing as many events as possible
 - automatic screening of events potentially related to anomalous behaviors
 - human analysis on a small number of screened events
- Applications
 - Current apps are either synthetic or small (sometimes both :-)



Automatic Event Screening: Diff against a reference (normal) system

- Given a system with performance anomalies, find a similar but normal system as a reference
- Collect comparable event traces from both systems and flag those that differ the most
- Challenges:
 - find a good reference (normal but similar to the anomalous system)
 - quantify and rank the difference of traced events and derived system metrics

Existence of a good reference: Anomaly Evolution over Software Versions



- Choose many runtime conditions (each condition specifies a concurrent I/O workload and an operating system configuration)
- For an anomalous system under a condition
 - Often (but not always), an earlier version at the same runtime condition does not exhibit anomaly
 - The earlier version can serve as a reference

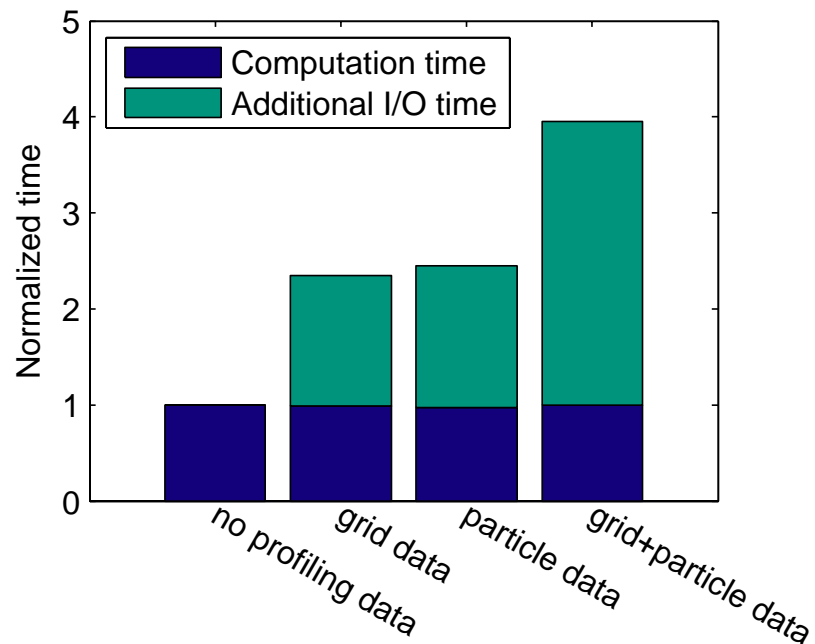


Existence of a good reference: Anomaly Characteristics

- For an anomalous system under a runtime condition
 - You can find some other conditions (using the same system version) that do not exhibit any anomaly
 - And often you can find “nearby” conditions (small variation in workload or small change in system configuration) that do not exhibit any anomaly
 - These “nearby” configurations can serve as references

Plan for Large-scale Application Study

- Collaboration with scientists at UR Lab of Laser Energetics
- OSIRIS: simulation to understand inertial confinement fusion
 - simulate particles in grids under compression laser beams
- Current performance: 85GFLOPS at 1024 processors



- Problem: output profiling data (mostly writes) is expensive
- Planned study:
 - understand its I/O behavior through distributed tracing
 - focus on writes (asynchronous)



Publications

- Supported Research

- [ICPP 2007] Multi-layer trace analysis and performance debugging for parallel I/O system
- [EuroSys 2007] Operating system-level balanced I/O prefetching for concurrent I/O system

- Related Research

- [USENIX 2007] VM data access tracing in virtual machine architecture
- [USENIX 2007] measurement of memory hardware errors on production systems

- Education

- [ACM SIGCSE 2006] Devise a new operating system course project that allows student to realistically explore performance issues in I/O systems